

Remarks

This REPLY is in response to the Office Action mailed December 12, 2008.

I. Summary of Examiner's Rejections

In the Office Action, Claims 50-63 and 71-75 were rejected under 35 U.S.C. 103(a) as being unpatentable over "Event-Oriented Dynamic Adaptation of Workflows: Model, Architecture, and Implementation" (hereinafter Muller) in view of "Web Service Orchestration" (HP, 1-2003, hereinafter Peltz).

II. Summary of Applicants' Amendments

No claims are amended by this REPLY. Reconsideration of the Application in view of the following remarks is respectfully requested.

III. Claim Rejections under 35 U.S.C. §103(a)

In the Office Action, Claims 50-63 and 71-75 were rejected under 35 U.S.C. 103(a) as being unpatentable over Muller in view of Peltz.

Applicant respectfully traverses the rejection of Claim 50. As currently presented, Claim 50 defines using a workflow language *in the form of annotations to the source code and the classes* of JAVA source code to enable developers that are used to working with Java to add workflow definitions. Such annotation would normally be ignored when the source code is processed, because annotations normally are not part of the program itself, and they normally have no direct effect on the operation of the source code they annotate. However, as defined by Claim 50, such annotations are read by the system and are used to create a workflow program, rather than being ignored.

Muller discloses an agent-based workflow management system that can deal with various failure situations. As disclosed therein, the AGENTWORK system supports the definition, the execution and, as its main contribution, the event-oriented and semi-automated dynamic adaption of workflows (page 1). Figure 1-2 of Muller (at page 5) illustrates a simplified architecture of a workflow management system, which includes a workflow definition. In the Office Action it was acknowledged that Muller does not explicitly address that the program source file includes a source code and classes therein, and workflow definition created using a workflow language that is specified in the form of annotations to the source code. However, it was asserted that Peltz discloses that a workflow definition is invoked using WSFL where WSFL

includes Java, i.e., program source file that includes source code and classes therein, and JavaDoc annotation tags in the workflow language that is specified in the form of annotations to the source code and classes.

Applicant respectfully traverses the rejection. In particular, applicant respectfully submits that Peltz does not disclose *a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes*. In the Office Action it was admitted that a workflow definition can be invoked using Web Services Flow Language (WSFL) (page 5 of Peltz); and that certain process logic can be written using Java program source which includes program source code and classes. However, it does not appear that Peltz discloses a workflow language that is specified in the form of annotations to the source code and the classes; rather, it is disclosed that a workflow definition can be invoked using WSFL or Java program source code. As such, Applicant respectfully submits that Peltz does not disclose that annotations to source code and classes can be used to add workflow constructs, in the manner defined by Claim 50.

Furthermore, it was asserted in the Office Action that JavaDoc annotation tags in the workflow language are specified in the form of annotations to the source code and classes (Office Action, page 6). For example, to start a conversation, a developer would include the tag, “@jws:conversation: phase=start” (page 10). Thus, while it appears that Peltz states that a conversation can be started using a JavaDoc annotation tag, the conversation defined therein is a way to identify a two-way communication, e.g., between one client application and one Web Service so that messages are always returned to the correct client. More specifically, the conversation described therein is a process that causes a response to a request to be returned to the correct requestor. Applicant respectfully submits that identifying a two-way communication is not the same as using a workflow language that is specified in the form of annotations to the source code and the classes, as is defined by Claim 50. Accordingly, since a workflow is a set of steps or actions that occur in a particular order, and in response to specific conditions, in order to perform a task Applicant respectfully submits that the statement in Peltz that a JavaDoc annotation tag can be used to start a conversation does not teach or suggest that a workflow definition can be created *using a workflow language that is specified in the form of annotations to source code and classes* of a program source file, as is defined by Claim 50.

In view of these comments, Applicant respectfully submits that Claim 50 is neither anticipated by, nor obvious in view of the cited references when considered alone or in combination, and reconsideration thereof is respectfully requested.

Claim 57 and 75

The comments provided above with respect to Claim 50 are hereby incorporated by reference. Claim 57, while independently patentable, recites limitations that, similarly to those described above with respect to Claim 50, are not taught, suggested nor otherwise rendered obvious by the cited references. For similar reasons as provided above with respect to Claim 50, Applicant respectfully submits that Claim 57 is likewise neither anticipated by, nor obvious in view of the cited references, when considered alone or in combination, and reconsideration thereof is respectfully requested.

Claims 51-56 and 58-63

Claims 51-56 and 58-63 depend from and include all of the features of Claims 50 and 57. Claims 51-56 and 58-63 are not addressed separately but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and the comments provided above. Reconsideration thereof is respectfully requested.

IV. Conclusion

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Applicants believe that no fee is due with this communication. However, the Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this reply, including any fee for extension of time which may be required.

Respectfully submitted,

Date: February 12, 2009

By: /Adam T. Hipp/
Adam T. Hipp
Reg. No. 60,334

Customer No. 80548
FLIESLER MEYER LLP
650 California Street, 14th Floor
San Francisco, California 94108
Telephone: (415) 362-3800